

**AD-A236 645**

GL-TR-90-0215



2

Ionospheric Characteristics for Archiving  
at the World Data Centers

Robert R. Gamache and Bodo W. Reinisch

University of Lowell  
Center for Atmospheric Research  
450 Aiken Street  
Lowell, Massachusetts 01854

December 1990

Scientific Report No. 3

Approved for public release; distribution unlimited.

GEOPHYSICS LABORATORY  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
HANSCOM AIR FORCE BASE, MA 01731-5000


DTIC  
ELECTE  
JUN 12 1991  
S B D


91 6 7 155

91-01650

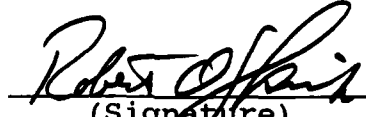


"This technical report has been reviewed and is approved for publication"

  
(Signature)  
JURGEN BUCHAU  
Contract Manager

  
(Signature)  
WILLIAM K. VICKERY  
Branch Chief

FOR THE COMMANDER

  
(Signature)  
ROBERT A. SKRIVANEK  
Division Director

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify GL/IMA, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it is returned.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ULRF-460/CAR			5. MONITORING ORGANIZATION REPORT NUMBER(S) GL-TR-90-0215		
6a. NAME OF PERFORMING ORGANIZATION University of Lowell		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Geophysics Laboratory		
6c. ADDRESS (City, State, and ZIP Code) Center for Atmospheric Research 450 Aiken Street Lowell, Massachusetts 01854			7b. ADDRESS (City, State, and ZIP Code) Hanscom Air Force Base Massachusetts 01731-5000		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Geophysics Laboratory		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-90-K-0029		
8c. ADDRESS (City, State, and ZIP Code) Hanscom Air Force Base Massachusetts 01731-5000			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			124417F	4643	08
11. TITLE (Include Security Classification) Ionospheric Characteristics for Archiving at the World Data Centers					
12. PERSONAL AUTHOR(S) Robert R. Gamache; Bodo W. Reinisch					
13a. TYPE OF REPORT Scientific No. 3		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1990 December	
15. PAGE COUNT 52					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Ionospheric Characteristics, World Data Center Archiving		
			IIWG Database for Uneven Time Rates		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>A database structure for archiving ionospheric characteristics at uneven data rates was developed at the July 1989 Ionospheric Informatics Working Group (IIWG) Lowell Workshop on Digital Ionogram Data Formats for World Data Center Archiving. This structure is proposed as a new URSI standard and is being employed by the World Data Center A for solar terrestrial physics for archiving characteristics. Here the database has been slightly refined for the application and programs written to generate these database files using as input Digisonde 256 ARTIST [Reinisch and Huang, 1983; Reinisch et al., 1983; Tang et al., 1989] data, post processed by the ULCAR ADEP [Zhang, 1989; Zhang et al., 1988] (ARTIST Data Editing Program) system. The characteristics program as well as supplemental programs developed for this task are described here. The new software will make it possible to archive the ionospheric characteristics from the Geophysics Laboratory high latitude Digisonde network, the AWS DISS and the international Digisonde networks, and other ionospheric sounding networks.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Jurgen Buchau			22b. TELEPHONE (Include Area Code)		22c. OFFICE SYMBOL GL/LIS

# TABLE OF CONTENTS

	Page
1. Introduction	1
2. Description of characteristics database	1
3. Implementation for ADEP Data	9
4. Supplemental Programs	14
5. Summary	14
6. References	16
APPENDIX A OUTPUT FROM RDBLOCK.FOR	18
APPENDIX B PROGRAM CHARS.FOR	43

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



## LIST OF TABLES

Table No.		Page
1	URSI IIWG Database Structure for Flexible Data Rates	2
2	List of Characteristics, URSI codes, and Dimensions	5
3	IIWG Codes for the Descriptive and Qualifying Fields of the Characteristics.	8
4	IIWG ADEP Structure as of January 31, 1990	10
5	ADEP edit flags.	13

## 1. Introduction

The archiving of ionospheric characteristics at uneven data rates was presented by Gamache and Reinisch [1989a] at the International Workshop on Digital Ionogram Data Formats for World Data Center Archiving [Gamache and Reinisch, 1989b] as part of the URSI Working Group G.4: Ionospheric Informatics. Because of the uneven time sampling, the database has records of varying sizes. Historically, this has created problems in both the creation of and in later utilization of the files. The IIWG proposed database avoids these problems by having the header records serve as a key to encoding the remainder of the data. The database is standard ASCII characters and is created, serviced, and used via FORTRAN programs. For a full description of the database and the philosophy behind its creation see Gamache and Reinisch 1898a. Table 1 gives the revised IIWG proposed structure of the database.

## 2. Description of characteristics database

Before describing the structure of the database, it is important to give some definitions to aid in the discussion. A database record, DR, is defined as the collection of all characteristics data for a month. The database is created by a FORTRAN program (this can also be done in C or any high level language) and each FORTRAN WRITE statement creates a FORTRAN record, FR, from which the database is comprised. For ease of use, transportability, editing, and printing, the FR length was set to 120 characters maximum. When a DR is created on a hard, floppy, or optical disk it is called a file and is a single entity. When created on a magnetic tape, the file is constructed in a number of blocks. Tape handling routines put a 0.6 inch gap between blocks. For magnetic tape the block size has been set by adjusting the block size to maximize the ratio of the inter-block gaps and data stored. On some systems the block size is governed by software that limits the counter to a four byte word yielding a maximum block size of 9999 bytes. For transportability, a

Table 1. URSI IIWG Database Structure for Flexible Data Rates

Data Group	FORTTRAN Record #	Format	Description
1	1	A30	Station Name
	1	A5	Station code
	1	I4	Meridian time used by station on records
	1	F5.1	Latitude N
	1	F5.1	Longitude E
	1	A10	Scaling type: Manual/Automatic
	1	A10	Data editing: Edited/Non-edited/Mixed
	1	A30	Ionosonde system name
2	2,3	30I4*	Year Month Number of days in the month, M Number of Characteristics Numbers of measurements total Numbers of measurements for each of the M days, $N_M$
	4,i	12A10*	List of characteristics
	i+1,j	12A10*	Dimensions
	j+1,k	60A2*	List of corresponding URSI codes
3	k+1,l •	20(3I2)* ••	The $N_M$ sample times Hh:Mm:Ss for each of the M days
	l+1,m •	24(I3,2A1)* ••	The $N_1$ values of characteristic 1 for day 1 •••repeated for each of the M days
	m+1	24(I3,2A1)	Hourly Medians for characteristic 1
	m+2	24(I2,2A1)	The counts for the hourly medians,Range
	m+3	24(I3,2A1)	Upper quartile
	m+4	24(I3,2A1)	Lower quartile
	m+5	24(I3,2A1)	Upper decile
	m+6	24(I3,2A1)	Lower decile
	m+7,n •	24(I3,2A1)* •	The $N_2$ values of characteristic 2 for day 1 • • etc.
	•	• •	repeated for each characteristic • • •

\* Format is repeated as many times as necessary to read/write the data.

Revised:11 April 1991.

block size of 9000 bytes was chosen. This, for example, is capable of storing two characteristics of routine hourly measurements within a block on magnetic tape.

The structure of the IIWG characteristics database for a database record is displayed in Table 1. A given DR is comprised of a number of FRs, the total number of FRs being determined by the number of days in the month, the number of measurements made each day, and the number of characteristics being archived. The database record is constructed with the data being in three groups. The first group is informative and encoding data. This starts with a FR containing the **Station Name**, (A30) format, where the data were recorded and the **Station Code** in (A5) format; the **meridian time** used by the station to indicate if time is recorded in UT or LT on the records is given next in (I4) format followed by the station coordinates, **Latitude N** and **Longitude E** both in (F5.1) format; next there are two (A10) format variables describing the **Scaling type**, this takes the value *Manual* or *Automatic*, and the **Data editing** variable which can be *Edited*, *Non-Edited*, or *Mixed*.; last in this FR is space for the **Ionosonde system name** in (A30) format.

FORTTRAN records two and three contain most of the information for encoding the DR. A repeating (30I4) format is used to write both records. They start with the **Year, Month, Number of days** in the month (NN), the **Number of characteristics** (Nchar), archived in this DR, the **Total number of measurements** reported for the DR, and the **Number of daily measurements** made for each of the NN days. FR number four to record *i* use a repeating format of (I2A10) to list the **Names** of the particular characteristics being archived. There are Nchar of them, hence the FRs are repeated as needed. For example if one were archiving only the critical frequencies foF2, foF1, and foE, Nchar would be three, and the characteristics list would be ' foF2' ' foF1' ' foE'. A list of the names of the characteristics, the units, and URSI codes taken from UAG23 (Piggott and Rawer, 1978) are given in Table 2; The URSI list has been enhanced with characteristics that are scaled by the Digisonde ARTIST [Tang et al. 1989]. FORTTRAN records *i+1* to *k* give the **Dimensions** corresponding to the characteristics list (see

Table 2), these are in (12A10) repeating format. The last FRs in the first group are for the **URSI codes** specified for each of the characteristics (see Table 2) and are written in (60A2) repeating format.

From the information in this first group one knows immediately how many data for the time or for each characteristic are to be read. From the number of measurements for each day the time data can be separated into the times for the individual days of the month and the measured characteristics can uniquely be associated with a given time on a given day.

The second group of FRs contain the measurement times for the month. With uneven time spacing the measurement times must be recorded to associate with the reported characteristics. This requires that hours, minutes, and seconds of each measurement be entered into the database. To conserve space, the times are written once per month and the reported characteristics are written to correspond to these times. FORTRAN records  $k+1$  to  $l$  are used for the measurement times. They are written in a (30(312)) repeating format corresponding to the **hours, minutes, and seconds, HHMMSS**, of the measurements. The number of FRs needed for this is determined by the data sampling rate for the month.

The third group of FRs contains the actual values of the characteristics and the corresponding hourly medians and statistics. The group is comprised of a number of FRs for each archived characteristic which are repeated for each characteristic. The order of the characteristics follows that given in the "List of characteristics". On a per characteristic basis, for each characteristic one has the  $N_1$  values of the characteristic for day 1 corresponding to the reported measurement times for day 1. These are followed by the values for day 2, day 3, ... for each of the NN days of the month. The **characteristics** are written in a repeating (24(I3,2A1)) format corresponding to the integer value (I3) of the characteristic and the qualifying and descriptive letters [see UAG 23].

The IIWG Workshop suggested the use of two slashes, //, in place of the qualifying and descriptive letters for monthly characteristics data that were autoscaled but not checked or "edited",

Table 2. List of Characteristics, URSI codes, and Dimensions

GROUP	CHARACTERISTIC		URSI	DIMENSION	REFERENCE	DEFINITION
	ARTIST					
	Name	#	Name	#	UAG23	
F2		1	foF2	00	.1 MHz	1.11 The ordinary wave critical frequency of the highest stratification in the F region
			fxF2	01	.1 MHz	1.11 The extraordinary wave critical frequency
			fzF2	02	.1 MHz	1.11 The z-mode wave critical frequency
	M(D)	3	M3000F2	03	.01 MHz	1.50 The maximum usable frequency at a defined distance divided by the critical frequency of that layer
	hpF2	12	h'F2	04	km	1.33 The minimum virtual height of the ordinary wave trace for the highest stable stratification in the F region
			hpF2	05	km	1.41 The virtual height of the ordinary wave mode at the frequency given by 0.834 of foF2 (or other 7.34)
			h'Ox	06	km	1.39 The virtual height of the x trace at foF2
	MUF(D)	4	MUF3000F2	07	.1 MHz	1.5C The standard transmission curve for 3000 km
			hc	08	km	1.42 The height of the maximum obtained by fitting a theoretical h'F curve for the parabola of best fit to the observed ordinary wave trace near foF2 and correcting for underlying ionization
			qc	09	km	7.34 Scale height
F1		2	foF1	10	.01 MHz	1.13 The ordinary wave F1 critical frequency
			fxF1	11	.01 MHz	1.13 The extraordinary wave F1 critical frequency
				12		not used
			M3000F1	13	.01 MHz	1.50 See Code 03
			h'F1	14	km	1.30 The minimum virtual height of reflection at a point where the trace is horizontal
				15		not used
	hpF	11	h'F	16	km	1.32 The minimum virtual height of the ordinary wave trace taken as a whole
			MUF3000F1	17	.1 MHz	1.5C See Code 07
				18		not used
				19		not used
E		9	foE	20	.01 MHz	1.14 The ordinary wave critical frequency of the lowest thick layer which causes a discontinuity
				21		not used
			foE2	22	.01 MHz	1.16 The critical frequency of an occulting thick layer which sometimes appears between the normal E and F1 layers
				23		not used
	hpE	13	h'E	24	km	1.34 The minimum virtual height of the normal E layer
				25		not used
			h'E2	26	km	1.36 The minimum virtual height of the E2 layer trace
				27		not used
				28		not used
				29		not used

Table 2..continued

GROUP	CHARACTERISTIC		DIMENSION	REFERENCE	DEFINITION
	ARTIST	URSI			
	Name	#	Name	#	
Es		6	foEs	30	.1 MHz 1.17 The highest ordinary wave frequency at which a mainly continuous Es trace is observed
			fxEs	31	.1 MHz 1.17 The highest extraordinary wave frequency at which a mainly continuous Es trace is observed
			fbEs	32	.1 MHz 1.18 The blanketing frequency of the Es layer
			ftEs	33	.1 MHz Top frequency Es any mode.
	hpEs	14	h'Es	34	km 1.35 The minimum height of the trace used to give foEs
				35	not used
			Type Es	36	7.26 A characterization of the shape of the Es trace
				37	not used
				38	not used
				39	not used
Other			foF1.5	40	.01 MHz 1.12 The ordinary wave critical frequency of the intermediate stratification between F1 and F2
				41	not used
		5	fmin	42	.1 MHz 1.19 The lowest frequency at which echo traces are observed on the ionogram
			M3000F1.5	43	.01 MHz 1.50 See Code O3
			h'F1.5	44	km 1.38 The minimum virtual height of the ordinary wave trace between foF1 and foF1.5 (equals h'F2 7.34)
				45	not used
				46	not used
			fm2	47	.1 MHz 1.14 The minimum frequency of the second order trace
			hm	48	km 7.34 The height of the maximum density of the F2 layer calculated by the Titheridge method
			fm3	49	.1 MHz 1.25 The minimum frequency of the third order trace
Spread F/Oblique			foI	50	.1 MHz 1.26 The top ordinary wave frequency of spread F traces
		10	fxI	51	.1 MHz 1.21 The top frequency of spread F traces
			fmI	52	.1 MHz 1.23 The lowest frequency of spread F traces
			M3000I	53	.01 MHz 1.50 See Code O3
			h'I	54	km 1.37 The minimum slant range of the spread F traces
				55	not used
				56	not used
			dfs	57	.1 MHz 1.22 The frequency spread of the scatter pattern
				58	7.34 Frequency range of spread fxI-foF2
				59	not used

Table 2..continued

GROUP	CHARACTERISTIC		DIMENSION	REFERENCE	DEFINITION			
	ARTIST	URSI						
	Name	#	Name	#	UAG23			
N(h)		30	fh'F2	60	.1 MHz	7.34	The frequency at which h'F2 is measured	
		29	fh'F	61	.1 MHz	7.34	The frequency at which h'F is measured	
				62			not used	
			h'mF1	63	km	7.34	The maximum virtual height in the o-mode F1 cusp	
			h1	64	km	7.34	True height at f1 Titheridge method	
			h2	65	km	7.34	True height at f2 Titheridge method	
			h3	66	km	7.34	True height at f3 Titheridge method	
			h4	67	km	7.34	True height at f4 Titheridge method	
			h5	68	km	7.34	True height at f5 Titheridge method	
			H	69	km	7.34	Effective scale height at hmF2 Titheridge method	
	T.E.C.			I2000	70	e/cm <sup>3</sup>	7.34	Ionospheric electron content Faraday technique
				I	71	e/cm <sup>3</sup>	7.34	Total electron content to geostationary satellite
			Ixxxx	72	e/cm <sup>3</sup>	7.34	Ionospheric electron content to height xxxx	
				73			not used	
				74			not used	
				75			not used	
				76			not used	
				77			not used	
				78			not used	
			T	79	e/cm <sup>3</sup>	7.34	Total sub-peak content Titheridge method	
AUTOMATIC		7	FMINF	80	.01 MHz		Minimum frequency of F trace (50 kHz increments) Equals fbEs when E present	
		8	FMINE	81	.01 MHz		Minimum frequency of E trace (50 kHz increments).	
		15	HOM	82	km		Parabolic E region peak height	
		16	YM	83	km		Parabolic E region semi-thickness	
		17	QF	84	km		Average range spread of F trace	
		18	QE	85	km		Average range spread of E trace	
		22	FF	86	.01 MHz		Frequency spread between fxF2 and fxI	
		23	FE	87	.01 MHz		As FF but considered beyond foE	
		25	fMUF3000	88	.01 MHz		MUF(D)/obliquity factor	
		26	h'MUF3000	89	km		Virtual height at fMUF	

i.e. where no quality control procedure has been applied. This code has been extended to consider data that have been edited but no descriptive or qualifying letters introduced. With two positions to fill and the use of a single or double slash there are four codes which can be defined. The first is no slashes implying the use of the descriptive or qualifying letters. The next is the use of two slashes which signifies no editing. The third choice is to put a slash in the first position followed by a blank. This is used to signify autoscaled data that have been edited but no descriptive or qualifying letters are used. The last possibility is a blank in the first position followed by the slash. This is not currently used thus it leaves the possibility for future extension of the code. The codes are summarized in Table 3.

Table 3. IIWG Codes for the Descriptive and Qualifying Fields of the Characteristics.

Symbolic Code	Description
Q D	Qualifying and descriptive letters used according to UAG #23A.
/ —	Autoscaled data, edited but no qualifying and descriptive letters used.
— /	No current meaning, for future extension.
/ /	Autoscaled data, no editing, no qualifying and descriptive letters used.

The actual values of the characteristics can be obtained by multiplying the integer value by the value found in the corresponding "dimension list" (group 2) of the database record (see Table 2). Thus a value of 86 reported for foF2 is multiplied by the dimension 0.1 MHz to give a foF2 value of 8.6 MHz.

Immediately following the characteristics data are the **hourly medians** given in a (24(I3,2A1)) format; the **counts** for the hourly medians and the **range** in (24(I2,I3)) format; the **upper quartiles** in a (24(I3,2A1)) format; the **lower quartiles** in a (24(I3,2A1)) format; then the **upper deciles** in a (24(I3,2A1)) format; and finally the **lower quartiles** again in a (24(I3,2A1)) format.

The above FRs are repeated for each characteristic given in the "characteristics list." This completes the DR, i.e. a month of characteristics data.

The reading and writing of the database records to magnetic tape or other mediums are accomplished by FORTRAN (or other high level languages) programs distributed with the database (see the Appendices). The uneven time spacing leads to DR lengths that change with the data, but as shown this is no problem given the structure of the database. This is an important factor when transportability is considered.

The advantages of this structure are: the flexibility of the resulting database allows data rates to change from month to month and/or from station to station. The database is ANSI standard and transportable and is easily maintained and utilized by FORTRAN routines provided. The structure is very flexible and most importantly does not lead to data being discarded.

### 3. Implementation for ADEP data

The creation of the characteristics database has been automated for ULCAR ADEP data. The ADEP data, a structured database developed by Gamache et. al. [1989c], was also presented at the Lowell Workshop. At the workshop it was suggested that this database be adopted as a standard for archiving both manual and autoscaled ionogram data. To accomplish this several enhancements were added to the structure. In Table 4, the suggested structure for the IIWG ADEP database is given; this structure has been programmed into the ADEP program. This is the third change in the structure of the ADEP database, this however has not been a problem because the database was created to be upwards compatible. Several programs exist to convert the original ADEP data (type 1, 1988) to the Standard ADEP Output (SAO) data (type 2, 1989), and now from the SAO format to the IIWG recommended format (type 3, 1990), without loss of any data. The Workshop's recommendation was that the ADEP structure for scaled ionogram data serve as a guide for other ionosonde systems.

Table 4. IIWG ADEP Structure as of January 31, 1990

<u>Code</u>	<u>Format</u>	<u>Description</u>
	2(40I3)	DATA FILE INDEX
1	N(16F7.3)	GEOPHYSICAL CONSTANTS
2	N(A120)	SYSTEM DESCRIPTION: Version numbers, etc.
3	N(120Z1)	IONOGRAM SOUNDING SETTINGS (PREFACE)
4	N(15F8.3)	SCALED IONOSPHERIC PARAMETERS
5	N(60I2)	ANALYSIS FLAGS
6	N(16F7.3)	DOPPLER TRANSLATION TABLE
		O-TRACE POINTS - F2 LAYER
7	N(40I3)	VIRTUAL HEIGHTS
8	N(40I3)	TRUE HEIGHTS
9	N(60I2)	AMPLITUDES
10	N(120I1)	DOPPLER NUMBER
11	N(20F6.3)	FREQUENCY TABLE
		O-TRACE POINTS - F1 LAYER
12	N(40I3)	VIRTUAL HEIGHTS
13	N(40I3)	TRUE HEIGHTS
14	N(60I2)	AMPLITUDES
15	N(120I1)	DOPPLER NUMBER
16	N(20F6.3)	FREQUENCY TABLE
		O-TRACE POINTS - E LAYER
17	N(40I3)	VIRTUAL HEIGHTS
18	N(40I3)	TRUE HEIGHTS
19	N(60I2)	AMPLITUDES
20	N(120I1)	DOPPLER NUMBER
21	N(20F6.3)	FREQUENCY TABLE
		X-TRACE POINTS - F2 LAYER
22	N(40I3)	VIRTUAL HEIGHTS
23	N(60I2)	AMPLITUDES
24	N(120I1)	DOPPLER NUMBER
25	N(20F6.3)	FREQUENCY TABLE
		X-TRACE POINTS - F1 LAYER
26	N(40I3)	VIRTUAL HEIGHTS
27	N(60I2)	AMPLITUDES
28	N(120I1)	DOPPLER NUMBER
29	N(20F6.3)	FREQUENCY TABLE
		X-TRACE POINTS - E LAYER
30	N(40I3)	VIRTUAL HEIGHTS
31	N(60I2)	AMPLITUDES
32	N(120I1)	DOPPLER NUMBER
33	N(20F6.3)	FREQUENCY TABLE
34	N(60I2)	MEDIAN AMPLITUDE OF F ECHO
35	N(60I2)	MEDIAN AMPLITUDE OF E ECHO
36	N(60I2)	MEDIAN AMPLITUDE OF ES ECHO
37	N(13E9.4E1)	TRUE HEIGHT F2 LAYER COEFFICIENTS
38	N(13E9.4E1)	TRUE HEIGHT F1 LAYER COEFFICIENTS
39	N(13E9.4E1)	TRUE HEIGHT E LAYER COEFFICIENTS
40	N(13E9.4E1)	VALLEY DESCRIPTION
41	N(120I1)	EDIT FLAGS

The monthly characteristics program was developed to read in one month of ADEP data and create the IIWG characteristics database. This is accomplished by reading data for one ionogram at a time and storing the measurement time and characteristics data in arrays. The program keeps track of the number of measurements made each day of the month, which changes with the measurement schedule. Currently the arrays in the program for this are dimensioned 31X300 supporting a full month at a 5 minute schedule (288 measurements/day). If more measurements are made the array boundaries must be increased accordingly.

The locations of the hourly measurements (hours 0 to 23) in the measurements/day list are found and stored for use in the median routine. This array keys the locations of the hourly characteristics in the larger monthly characteristics arrays. Programming in this fashion is very memory intensive, however it is logical and produces a robust and error free algorithm. For example, the data supplied to the algorithm need not be ordered in time for the algorithm to work properly. This is important because a month of ADEP data, several megabytes, seldom exist as a file complete for the month but must be collated into a single file. Errors made in collating do not affect the characteristics program. Thus the results are independent of how the data for the month were put together.

The number of FRs necessary for archiving each characteristic is variable from month to month depending on the measurement schedule. The program makes all the necessary provisions and the key information is written into the first group of FRs for future encoding. In this way relatively little space is used for null data compared to the previous URSI characteristics files [Rodger 1984].

The program CHARS.FOR was written in FORTRAN77 and will run on any system supporting this language. It was developed on an IBM-PC/AT compatible system. Because of the large arrays in the program, COMMON blocks were constructed in groups each with less than 64K requirements. Data not in common blocks do not exceed 64K, however for safety the program was compiled using Microsoft FORTRAN option /Gt3000 which partitions this data so that the 64K

constraint is never exceeded. It is recommended that similar options be used on other systems when necessary.

The program was tested using several months of simulated ADEP data. The simulated data was constructed so that the value assigned to a characteristic was related to the measurement time. For example, the critical frequency of the F2 layer is defined by

$$foF2 = \text{Hour} + 2. + \text{Day}/10. \quad (1)$$

where Hour is the hour (0 to 23) and Day is the day of the month (1 to 31). This allows the resulting characteristics files and median, quartile, etc. results to be thoroughly understood and checked. The program was tested with input files that varied the number of measurements per day from 24 to 288. In the tests, the characteristics and the medians, quartiles, deciles, counts and ranges were checked and found to be correct.

In additional tests, the days of the month were scrambled and the characteristics files generated again. The order of the measurement times and characteristics were unchanged compared to the previous file and the medians, quartiles, etc. were unchanged by the reordering of the days. A similar test was done where in addition to scrambling days, the measurement times within each day were scrambled. The measurement times and characteristics in the measurements/day lists had a different order but they had a one to one correspondences in the FRs. Of course the hourly measurement indication vectors were very different but the medians, quartiles, counts, etc. were unchanged for the month. It is not suggested that ADEP data files be created without time ordering, however if this does happen it will not affect the final database record.

Appropriate coding has been added as edit flag information to the IIWG ADEP format, Table 4 #41. For these flags a 0 implies autoscaling and a 1 indicates the corresponding data has been manually edited. The list of data for the edit flags is shown in Table 5. The flag position is from the position of the scaled characteristic in the ARTIST and ADEP list with the addition of flags for the traces and true height. A program was written to take the standard ADEP output files, type 2, and add this information to the database to

TABLE 5. ADEP edit flags.

Edit flag Position	Scaled Characteristics	Description
1	foF2	F2 layer critical frequency calculated by hyperbolic fit
2	foF1	F1 layer critical frequency
3	M(D)	M-factor, $MUF(D)/foF2$ , for distance D
4	MUF(D)	Maximum usable frequency for distance D
5	fmin	Minimum frequency for E or F echoes
6	foEs	Es layer critical frequency
7	fminF	Minimum frequency of F-trace
8	fminE	Minimum frequency of E-trace
9	foE	E layer critical frequency
10	fxI	Maximum frequency of F-trace
11	h'F	Minimum virtual height of F trace
12	h'F2	Minimum virtual height of F2 trace
13	h'E	Minimum virtual height of E trace
14	h'Es	Minimum virtual height of Es layer
15	HOM	Peak of E layer using parabolic model
16	Ym	Corresponding half thickness of E layer
17	QF	Average range spread of F-trace
18	QE	Average range spread of E-trace
19	Down F2	Lowering of F-trace maximum to leading edge
20	Down E	Lowering of E-trace maximum to leading edge
21	Down Es	Lowering of Es-trace maximum to leading edge
22	FF	Frequency spread between $fxF2$ and $fxI$
23	FE	As FF but considered beyond foE
24	D	Distance used for MUF calculation <sup>†</sup>
25	fMUF(D)	$MUF(D)/\text{obliquity factor}^*$
26	h'MUF(D)	Virtual height at fMUF
27	foF2c	correction to add to foF2 to get actual foF2
28	foEp	Predicted foE
29	f(h'F)	Frequency at which hminF occurs
30	f(h'F2)	Frequency at which hminF2 occurs
31	foF1p	Predicted foF1
32	Zmax	Peak height F2 layer
50	F2 trace	F2 trace points were edited
51	F1 trace	F1 trace points were edited
52	E trace	E trace points were edited
53	true height	true height was recalculated with edited traces
54	Es trace	Es trace points were edited

<sup>†</sup> Normally 3000 km

\* Obliquity factor(h', D) is the ratio of frequencies for vertical and oblique propagation to distance D with virtual height h'.

produce the type 3 files, IIWG ADEP database files. With this as input the addition of slashes when necessary is fully automated.

We have used the term "autoscaled data" in the discussion since our experience is based on the Digisonde ARTIST- scaled ionogram data. All definitions apply, however, as well to manually or semi-automatically digitized data.

#### 4. Supplemental Programs

In addition to the FORTRAN program that creates the database records, an additional program was written to read the database records and write out the data along with some descriptive information. This program can also be used as a seed program to encode the characteristics database for other applications and studies involving the database files, allowing lesser experienced programmers quick access to the data. Like the database, the output files are written with a maximum record length of 120 characters allowing a study of the data. Currently fourteen characteristics are considered in the ULCAR version of the CHARS.FOR program: foF2, foF1, Mfac, MUF, fmin, foEs, fmnF, fmnE, foE, fxI, hpF, hpF2, hpE, hpEs. Definitions of these characteristics are given in Tables 2 and 4.

#### 5. Summary

Programs have been written to produce the monthly characteristic database files for uneven time sampling as recommended by the IIWG. Supplemental programs have been created to convert the original ADEP files (type 1, 1988) to the standard ADEP output files (type 2, 1989) and from these to the IIWG ADEP format (type 3, 1990). The ULCAR ADEP program now generates files in the IIWG ADEP format. Any monthly characteristics file in this format can be input to the characteristics program. The characteristics program, CHARS.FOR is given in Appendix A. A program to read and encode the characteristics database file is given in Appendix B. This program can be easily

modified to read the characteristics data for use in studies and other applications.

From these programs, ionosonde data collected from the worldwide Digisonde network [Reinisch 1986, Tang et. al. 1990] and from other networks or sounders using the IIWG ADEP format can quickly and confidently be used to generate ionospheric characteristics files for archiving. The later use of these files is easily accomplished (see Appendix B).

## 6. References

Gamache R. R. and B. W. Reinisch, "Ionogram Characteristics at Uneven Data Rates," Presented at URSI Working Group G.4 Ionospheric Informatics International Workshop, July, 1989., University of Lowell Center for Atmospheric Research, 1989a.

Gamache R. R. and B. W. Reinisch, Proceedings from the International Workshop on "Digital Ionogram Data Formats for World Data Center Archiving," University of Lowell Center for Atmospheric Research, November 1989b.

Gamache R. R., T. Bullett, Z. Zhang, B. W. Reinisch, and W. T. Kersey, "ADEP Database Report," University of Lowell Center for Atmospheric Research, 1989.

Piggott W. R. and K. Rawer, Editors, U.R.S.I. Handbook of Ionogram Interpretation and Reduction, World Data Center A for Solar-Terrestrial Physics Report UAG-23, and 23A, Boulder CO (1978).

Reinisch, B. W., "New Techniques in Ground-Based Ionospheric Sounding and Studies," Radio Science, 21, No. 3, pp. 331-341, May-June 1986.

Reinisch B. W., R. R. Gamache, J. S. Tang, and D. F. Kitrosser, "Automatic Real Time Ionogram Scaler with True Height Analysis - ARTIST," Scientific Report No. 7, AFGL-TR-83-0209, 1983. ADA135174

Reinisch B.W. and X. Huang, "Automatic Calculation of Electron Density Profiles from Digital Ionograms. 3. Processing of Bottomside Ionograms," Radio Sci. 18, 477-492 (1983).

Rodger A. S. , Ed. INAG Ionospheric Station Information Bulletin No. 45, Nov. 1984; "Guide to the World Data Center Systems," Part 2: Ionosphere, International Council of Scientific Unions, July 1989.

Tang J., C. G. Dozois, and R. R. Gamache, "ARTIST Tape Output Formats," Scientific Report No. XX, to be published, 1990.

Tang J., R. R. Gamache, and B. W. Reinisch, "Progress on ARTIST Improvements, "Scientific Report No. 14, GL-TR-89-0185, 1989.  
ADA219799

Zhang Z., "ADEP Editors User's Manual," University of Lowell Center for Atmospheric Research, August 1989

Zhang Z., T. Bullett, and B. W. Reinisch, "Modern Display Techniques for Ionospheric Parameters," University of Lowell Center for Atmospheric Research, March 1988.

APPENDIX A  
PROGRAM CHARS.FOR

```

c...compile with fl/Gt3000
$Storage: 2
c
  PROGRAM Chars
C
c...written by Bob Gamache
C    Read ADEP data for a month and create characteristics files
C    in IIWG Workshop format.
c...July 10, 1989
c...last modified December 13, 1990 by Bob Gamache
c
c...file 10 = Standard ADEP output for one month of data (1989
Format)
c...file 3  = IIWG recommended format monthly characteristics data
c...        (9000 byte blocks)
c...file 4  = informative data for the month - check feature
c
C
  parameter(N=31, M=288)
c
  Integer*2
cfoF2,cfoF1,cM3000,cMUF,cfmin,cfoEs,cfminF,cfminE,cfoE,
  + cfxI,chnF,chnF2,chnE,chnEs
c
  character*1
qfoF2,qMUF,qfoF1,qM3000,qfmin,qfoEs,qfminF,qfminE,
  + qfoE,qfxI,qhpF,qhpF2,qhpE,qhpEs
c
  Integer*4 Itimes,Ihr
c
  INTEGER*2 IDFI(80),IPREF(60),IAF(20),IOTF(400),IOTHF(400),
  +
IOAF(400),IODF(400),IOTF1(150),IOTHF1(150),IOAF1(150),IODF1(150
)
  +
,IOTE(150),IOTHE(150),IOAE(150),IODE(150),IXTF(400),IXAF(400),
  +
IXDF(400),IXTF1(150),IXAF1(150),IXDF1(150),IXTE(150),IXAE(150),
  +
  IXDE(150),MEDF(20),MEDE(20),MEDES(20),IedF(120)
  REAL
SCALED(45),GCONST(5),DTT(16),FOF(400),FOF1(150),FOE(150),
  + FXF(400),FXF1(150),FXE(150),THF2(20),THF1(20),THE(20),
  + THMON(20),THVAL(20)

```

```

Character*120 Sysdes
C
CHARACTER*40 IFILE,Iout
CHARACTER*30 Station,System
CHARACTER*10 Scale(2),Edit(3)
CHARACTER*5 cStat
CHARACTER*10 char(80),units(80),UT
character*2 code(80)
character*1 ql,aQL
LOGICAL EOF
c...characteristics data arrays
Common/Chard1/cfoF2(N,M),cMUF(N,M),cfoF1(N,M)
Common/chard2/cM3000(N,M),cfmin(N,M),cfoEs(N,M)
Common/chard3/cfminF(N,M),cfminE(N,M),cfoE(N,M)
Common/chard4/cfxI(N,M),chpF(N,M),chpF2(N,M)
Common/chard5/chpE(N,M),chpEs(N,M)
c
c...characteristics data array qualifiers
Common/QL1/qfoF2(N,M),qMUF(N,M),qfoF1(N,M)
Common/QL2/qM3000(N,M),qfmin(N,M),qfoEs(N,M)
Common/QL3/qfminF(N,M),qfminE(N,M),qfoE(N,M)
Common/QL4/qfxI(N,M),qhpF(N,M),qhpF2(N,M)
Common/QL5/qhpE(N,M),qhpEs(N,M)
c
c...times and measurements
Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
Common/time4/ltimes(N,M)
c
dimension aQL(0:1)
c
c...list of characteristics followed by Units and URSI codes
data char/ foF2,' foF1,' M3000F2,' MUF3000F2',
+ ' fmin',' foEs',' fmnF',' fmnE',
+ ' foE',' fxI',' h"F',' h"F2',
+ ' h"E',' h"Es', 66*' '/'
c
data units/ 0.1 MHz',' 0.01 MHz',' 0.01',' 0.1 MHz',
+ ' 0.1 MHz',' 0.1 MHz',' 0.01 MHz',' 0.01 MHz'
+ ' 0.01 MHz',' 0.1 MHz',' 1.0 km',' 1.0 km',
+ ' 1.0 km',' 1.0 km', 66*' '/'
c
data code/'00','10','03','07','42','30','80','81','20','51','16',
+ '04','24','34','66*' '/'

```

```

C
    Scale(1) = '    Manual'
    Scale(2) = ' Automatic'

c
    Edit(1) = '    Edited'
    Edit(2) = 'Non-Edited'
    Edit(3) = '    Mixed'

c
c...qualifying letters from ADEP edit flage
    aQL(1) = ' '
    aQL(0) = '/'
    IUNIT = 10
    n19 = 19
c...# characteristics
    ncar = 14

c
    K = 0
c...DIGISONDE operates in UT
    UT = '0000'

c
c...preset arrays
    call preset

c
    WRITE(*,*) 'Enter name of input file.'
    READ(*,'(A)') IFILE
    OPEN(UNIT=IUNIT,FILE=IFILE,FORM='FORMATTED',
+ ACCESS='SEQUENTIAL')

c
    WRITE(*,*) 'Enter name of output file.'
    READ(*,'(A)') Iout
    OPEN(3,FILE=Iout,FORM='FORMATTED',
+ ACCESS='SEQUENTIAL', RECL=9000)

c
    WRITE(*,*) 'Enter ionosonde station name, A30'
    read(*,'(a30)') station

c
3  WRITE(*,*) 'Scaling Type'
    Write(*,*) '      - Enter 1 for Manual scaling '
    Write(*,*) '      - Enter 2 for Automatic scaling '
    read(*,*) Iscal
    if(Iscal .ne. 1 .and. Iscal .ne. 2 ) then
        Write(*,*) 'incorrect entry, try again'
        go to 3

```

```

endif
c
5  WRITE(*,*) 'Data Editing '
   Write(*,*) '      - Enter 1 for Edited '
   Write(*,*) '      - Enter 2 for Not-edited'
   Write(*,*) '      - Enter 3 for Mixed'
   read(*,*) Iedit
   if(Iedit .ne. 1 .and. Iedit .ne. 2 .and. Iedit .ne. 3) then
       Write(*,*) 'incorrect entry, try again'
       go to 5
   endif
c
   WRITE(*,*) 'Enter ionosonde system name, A30'
   read(*, '(a30)') System
c
   ql = ''
   if(Iedit .eq. 2) ql = '/'
C
15  CONTINUE
    K = K + 1
    CALL
    RDADEP(EOF,IUNIT,IDFI,IPREF,SCALED,GCONST,IAF,DTT,IOTF,
    +
    IOTHF,IOAF,IODF,FOF,IOTF1,IOTHF1,IOAF1,IODF1,FOF1,IOTE,IOTHE,
    +
    IOAE,IODE,FOE,IXTF,IXAF,IXDF,FXF,IXTF1,IXAF1,IXDF1,FXF1,IXTE,
    +
    IXAE,IXDE,FXE,MEDF,MEDE,MEDES,THF2,THF1,THE,THVAL,IedF,Sysdes)
C
    if(EOF) go to 25
c
c...find day and time
    Iyr = IPREF(1)*10 + IPREF(2)
c...add century to year
    Iyr = n19*100 + Iyr
    Iday = IPREF(3)*100 + IPREF(4)*10 + IPREF(5)
    Ihr = IPREF(6)*10 + IPREF(7)
    Min = IPREF(8)*10 + IPREF(9)
    Isec = IPREF(10)*10 + IPREF(11)
c...for debugging
    write(11, '(5x,a,4i5)') 'ionogram #, yr,day,hour:',K,Iyr,Iday,Ihr
c
    if(Iday.gt.366 .or. Ihr.gt.23) go to 15

```

```

c
c...month, day of the month, and number of days in the month
    call dmonth(Iyr, Iday, Month, Mday)
    if(Mday.gt.31) go to 15
c
c...for hourly values the day starts at 23:54
    if(Ihr .eq. 23 .and. Min .gt. 54) then
        Mday = Mday +1
c...do not exceed days in the month
        if(Mday .gt. NN) go to 15
    endif
c
c...increment counter for number of measurements per day
    md(Mday) = md(Mday) + 1
    if(md(Mday) .gt. 288)
        + write(*,*) ' Warning md(',Mday,')=)',md(Mday)
c...time of measurement
    Itimes(Mday,md(mday)) = Ihr*10000 + Min*100 + Isec
c
c...store characteristic data in arrays
    cfoF2(Mday,md(Mday)) = ((scaled(1)+.001)*10)
    qfoF2(Mday,md(Mday)) = aQL(IedF(1))
    cfoF1(Mday,md(Mday)) = ((scaled(2)+.001)*10)
    qfoF1(Mday,md(Mday)) = aQL(IedF(2))
    if(scaled(24).eq.3000.) then
        if(scaled(3) .lt. 50.) then
            cM3000(Mday,md(Mday)) = ((scaled(3)+.0001)*100)
        else
            cM3000(Mday,md(Mday)) = 9999
        endif
        cMUF(Mday,md(Mday)) = ((scaled(4)+.001)*10)
    else
        cM3000(Mday,md(Mday)) = 9999
        cMUF(Mday,md(Mday)) = 9999
    endif
    qM3000(Mday,md(Mday)) = aQL(IedF(3))
    qMUF(Mday,md(Mday)) = aQL(IedF(4))
    cfmin(Mday,md(Mday)) = ((scaled(5)+.001)*10)
    qfmin(Mday,md(Mday)) = aQL(IedF(5))
    cfoEs(Mday,md(Mday)) = ((scaled(6)+.001)*10)
    qfoEs(Mday,md(Mday)) = aQL(IedF(6))
    cfminF(Mday,md(Mday)) = ((scaled(7)+.001)*10)
    qfminF(Mday,md(Mday)) = aQL(IedF(7))

```

```

    cfminE(Mday,md(Mday)) = ((scaled(8)+.001)*10)
    qfminE(Mday,md(Mday)) = aQL(IedF(8))
    cfoE(Mday,md(Mday)) = ((scaled(9)+.001)*10)
    qfoE(Mday,md(Mday)) = aQL(IedF(9))
    cfxI(Mday,md(Mday)) = ((scaled(10)+.001)*10)
    qfxI(Mday,md(Mday)) = aQL(IedF(10))
    chpF(Mday,md(Mday)) = ((scaled(11)+.01)*1)
    qhpF(Mday,md(Mday)) = aQL(IedF(11))
    chpF2(Mday,md(Mday)) = ((scaled(12)+.01)*1)
    qhpF2(Mday,md(Mday)) = aQL(IedF(12))
    chpE(Mday,md(Mday)) = ((scaled(13)+.01)*1)
    qhpE(Mday,md(Mday)) = aQL(IedF(13))
    chpEs(Mday,md(Mday)) = ((scaled(14)+.01)*1)
    qhpEs(Mday,md(Mday)) = aQL(IedF(14))
c
    go to 15
c
    25 continue
c
c...station number, latitude N, longitude E
    iScode = int(GCONST(5))
    Rlat = GCONST(3)
    Rlong = GCONST(4)
    Write(cStat,'(2x,i3) ) iScode
c
c...calculate hourly values for medians
    call hourly
c
    write(4,*) 'measurements/day'
    write(4,'(31I4,/)') md
c...find total number of measurements
    nmt = 0
    do 30 L=1,NN
    30 nmt = nmt + md(L)
c
c...calculate the number of days to write into a block
    call blocker
c
c...output first block
    write(3,'(A30,A5,A4,2F5.1,2A10,A30)') station,cStat,UT,Rlat,
+ Rlong,Scale(Iscal),Edit(Iedit),System
    write(3,'(30I4)') 1yr,Month,NN,ncar,nmt,md
    write(3,'(12A10)') (Char(L),L=1,ncar)

```

```

        write(3,'(12A10)') (units(L),L=1,ncar)
        write(3,'(60A2)') (code(L),L=1,ncar)
c
c...output measurement times..depends on the number of blocks
    call outime
c
c...compute mendian points
    call findMed(cfoF2, qfoF2, Medql, 900)
    call output(cfoF2, qfoF2, Medql, 900)
c
    call findMed(cfoF1, qfoF1, Medql, 900)
    call output(cfoF1, qfoF1, Medql, 900)
c
    call findMed(cM3000, qM3000, Medql, 900)
    call output(cM3000, qM3000, Medql, 900)
c
    call findMed(cMUF, qMUF, Medql, 900)
    call output(cMUF, qMUF, Medql, 900)
c
    call findMed(cfmin, qfmin, Medql, 900)
    call output(cfmin, qfmin, Medql, 900)
c
    call findMed(cfoEs, qfoEs, Medql, 900)
    call output(cfoEs, qfoEs, Medql, 900)
c
    call findMed(cfminf, qfminf, Medql, 900)
    call output(cfminf, qfminf, Medql, 900)
c
    call findMed(cfminE, qfminE, Medql, 900)
    call output(cfminE, qfminE, Medql, 900)
c
    call findMed(cfoE, qfoE, Medql, 900)
    call output(cfoE, qfoE, Medql, 900)
c
    call findMed(cfxI, qfxI, Medql, 900)
    call output(cfxI, qfxI, Medql, 900)
c
    call findMed(chpF, qhpF, Medql, 900)
    call output(chpF, qhpF, Medql, 900)
c
    call findMed(chpF2, qhpF2, Medql, 900)
    call output(chpF2, qhpF2, Medql, 900)
c

```

```

        call findMed(chpE, qhpE, Medql, 900)
        call output(chpE, qhpE, Medql, 900)
c
        call findMed(chpEs, qhpEs, Medql, 900)
        call output(chpEs, qhpEs, Medql, 900)
c
END
c
c*****
        subroutine output(Icharx, ql, dl, Medql, Ilimit)
c
        parameter(N=31, M=288)
c
        character*5 A5out(24)
        character*70 A70
        character*1 ql, qq, qa
        character*1 q1, d1
c
c...times and measurements
        Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
c
        common/median/Med(24), Icount(24), nUQ(24), LQ(24), nUD(24),
+ LD(24), IRange(24)
c
        dimension Icharx(N,M), ql(N,M)
c
        qa = '/'
c...output characteristic, medians, counts, upper/lower quartiles,
c...deciles and range
c
c...set qualifying and descriptive letters for medians
        if(Medql.eq.1) then
            ql = '/'
            dl = ' '
        else
            ql = '/'
            dl = '/'
        endif
c...output characteristics
c
        call blank(A5out)
        ns = 0
c...loop number of days

```

```

do 10 nd = 1, NN
c...loop measuremets per day
do 9 I = 1, md(nd)
ns = ns + 1
c
    if(Icharx(nd,I) .gt. Ilimit .or.
+      Icharx(nd,I) .eq. 0) then
        A5out(ns) = ' '
    else
        qq = qldl(nd,I)
        write(A5out(ns),'(I3,2A1)') Icharx(nd,I),qq,qq
    endif
c...when full output the start again
    if(ns .eq. 24) then
        write(3,'(24A5)') (A5out(kk),kk=1,ns)
        ns = 0
    endif
c
    9 continue
    10 continue
c
c...write out if there is any remaining data
    if(ns .gt. 0) write(3,'(24A5)') (A5out(kk),kk=1,ns)
c
c...medians
do 15 L = 1, 24
    write(A5out(L),'(I3,2A1)') Med(L),ql,dl
15    continue
    write(3,'(24A5)') (A5out(L),L=1,24)
c
c...median counts and Range
do 20 L = 1, 24
    write(A5out(L),'(I2,I3)') Icount(L),IRange(L)
20    continue
    write(3,'(24A5)') (A5out(L),L=1,24)
c
c...Upper Quartiles
do 25 L = 1, 24
    write(A5out(L),'(I3,2A1)') nUQ(L),ql,dl
25    continue
    write(3,'(24A5)') (A5out(L),L=1,24)
c
c...Lower Quartiles

```

```

        do 30 L = 1, 24
            write(A5out(L),'(I3,2A1)') LQ(L),ql,dl
30         continue
            write(3,'(24A5)') (A5out(L),L=1,24)
c
c...Upper Deciles
        do 35 L = 1, 24
            write(A5out(L),'(I3,2A1)') nUD(L),ql,dl
35         continue
            write(3,'(24A5)') (A5out(L),L=1,24)
c
c...Lower Deciles
        do 40 L = 1, 24
            write(A5out(L),'(I3,2A1)') LD(L),ql,dl
40         continue
            write(3,'(24A5)') (A5out(L),L=1,24)
c
50    continue
        return
        end
c
c*****
        subroutine outime
c
        parameter(N=31, M=288)
c
        character*6 A6out(20)
c
c...times and measurements
        Integer*4 Itimes
        Common/time4/Itimes(N,M)
        Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
c
c...output measurement times
c
        call blank(A6out)
        ns = 0
c...loop number of days
        do 10 nd = 1, NN
c...loop measuremnets per day
        do 10 I = 1, md(nd)
            ns = ns + 1
            write(A6out(ns),'(I6)') Itimes(nd,I)

```

```

c
    if(ns .eq. 20)then
        write(3,'(20A6)') (A6out(kk),kk=1,ns)
        ns = 0
    endif
10    continue
c
c...write out if there is any remaining data
    if(ns .gt. 0) write(3,'(20A6)') (A6out(kk),kk=1,ns)
c
    return
end
c
c*****
c      subroutine hourly
c*****
C
    parameter(N=31, M=288)
c
    Integer*4 Itimes,Ihr,min
c
c...times and measurements
    Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
    Common/time4/Itimes(N,M)
    dimension Jdel(N,M)
c
    do 5 I=1,31
        do 5 J=1,24
            5    Jdel(I,J) = 10
c
c...set hourly times
c...loop days -- I
        do 50 I=1, NN
c...loop measurements/day -- J
            do 49 J=1, md(I)
                Jmk = J
c
                Ihr = Itimes(I,J)/10000
                Min = (Itimes(I,J) - Ihr*10000)/100
c
                if(Min .gt. 54) then
                    Mk = Ihr + 1
                    Id = I

```

```

        Idel = 60 - Min
        go to 20
    endif
c
    if(Min .lt. 6) then
        Mk = Ihr
        if(Mk.eq.0) Mk = 24
        Id = I
        Idel = Min
        go to 20
    endif
c
c...not in proper time zone
    go to 49
c
c...hourly value check for closest value
    20 continue
        if(Ihv(Id,Mk) .eq. 0) then
c...set initial value
            Ihv(Id,Mk) = Jmk
            Jdel(Id,Mk) = Idel
        else
c...compare to previous value
            if(Idel .le. Jdel(Id,Mk)) then
                Ihv(Id,Mk) = Jmk
                Jdel(Id,Mk) = Idel
            endif
        endif
c
    49 continue
    50 continue
c
        write(4,('/',2x,24I3')) ((Ihv(I,J),J=1,24),I=1,31)
        return
    end
c
c*****
    subroutine Blank(AA)
c
        character*120 AA
        ns = 1
        do 10 I = 1, 12
            AA(ns:ns+9)='

```

```

10 ns = ns + 10
   return
   end
c
c*****
      subroutine findMed(Icharx, qcharx, Medql, Ilimit)
c
      parameter(N=31, M=288)
c
c...times and measurements
      Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
c
c...find medians, counts, upper/lower quartiles-deciles and range
      character*1 qcharx
c
      common/median/Med(24), Icount(24), nUQ(24), LQ(24), nUD(24),
+ LD(24), IRange(24)
      dimension Icharx(N,M), qcharx(N,M), Ival(N),
+ lc1(31),lc2(31),mc1(31),mc2(31)
c
c...lc1, lc2, mc1, and mc2 are the URSI location vectors for the upper
c...and lower quartiles (see UAG-23 p185)
      data lc1/1,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,7,7,7,
+ 7,8,8/
      data lc2/1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,7,7,7,7,8,
+ 8,8,8/
      data mc1/1,1,2,3,4,5,6,6,7,8,9,9,10,11,12,12,13,14,15,15,16,17,
+ 18,18,19,20,21,21,22,23,24/
      data mc2/1,1,3,4,5,5,6,7,8,9,9,10,11,11,12,13,14,14,15,16,17,17,
+ 18,19,20,20,21,22,23,23,24/
c
c...preset
      do 2 I=1,24
      Med(I) = 0
      Icount(I) = 0
      Irange(I) = 0
      nUQ(I) = 0
      LQ(I) = 0
      nUd(I) = 0
      LD(I) = 0
      2 continue
      Medql = 1
c

```

```

c...loop over hours
  do 50 I = 1, 24
c
    do 3 J=1,31
      Ival(J) = 0
    3 continue
c
c...get hourly value for each day
  nc = 0
c
  do 45 J = 1, NN
    Iloc = Ihv(J,I)
    if(Iloc .ne. 0) then
      if(Icharx(J,Iloc) .lt. Ilimit .and.
+       Icharx(J,Iloc) .ne. 0) then
        nc = nc + 1
        Ival(nc) = Icharx(J,Iloc)
        if(qcharx(J,Iloc) .eq. '/') Medql = 0
      endif
    endif
  45 continue
c
c
c...set counter
  Icount(I) = nc
c
  if(nc .eq. 0) go to 50
c...order the values
  call Order(Ival, nc)
c
c...nc=1
  if(nc .eq. 1) then
    Med(I) = Ival(1)
    go to 50
  endif
c
c...nc>2
  if(nc .ge. 2) then
c
c...median is at nc/2
    nc2 = nc/2
    if(Mod(nc,2) .eq. 0) then
      Med(i) = (Ival(nc2) + Ival(nc2+1)) / 2

```

```

        else
            Med(I) = Ival(nc2+1)
        endif
c
c...quartiles
        if(nc .ge. 3) then
            nUQ(I) = (Ival(mc1(nc)) + Ival(mc2(nc)) )/2
            LQ(I) = (Ival(lc1(nc)) + Ival(lc2(nc)) )/2
            IRange(I) = nUQ(I) - LQ(I)
        endif
c
c...deciles
        if(nc .ge. 10) then
            nc10 = nc/10
            nUD(I) = Ival(nc-nc10+1)
            LD(I) = Ival(nc10)
        endif
c
c...nc>2
        endif
c
50 continue
    return
end
c
c*****
subroutine Order(Kval, nc)
c
c...sort integer array Kval in assending order
    dimension Kval(31), Idum(31)
c
c...there are nc values
c
    Kount = nc
    do 100 I = 1, nc
        max = 0
        ip = 0
c...find max value
        do 50 J = 1, nc
            if(Kval(J) .ge. Max) then
                ip = J
                Max = Kval(J)
            endif

```

```

50 continue
   Idum(Kount) = Max
   Kval(ip) = 0
   Kount = Kount - 1
100 continue
c
c...put ordered array back into Kval
   do 150 K = 1, nc
150 Kval(K) = Idum(K)
c
   return
end
c
c*****
c      subroutine dmonth(Iyr, Iday, Month, Mday)
c
c      parameter(N=31, M=288)
c
c...times and measurements
      Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
c
c...from Iday calculate month, day of the month, and number
c...of days in the month.
c
c...Mth      first day of each month (Feb start)
c...Leap     leap year correction factor
c...Iyr      year (I2) e.g. 83 corresponds to 1983
c...Month     local month
c...Iday     local day (1-365)
c...Mday     local day (1-31)
c...Nmonth   # of days for each month
c...NN       # of days in the current month
c
      dimension Mth(12),Nmonth(12)
c
      data Mth/32,60,91,121,152,182,213,244,274,305,335,366/
      data Nmonth/31,28,31,30,31,30,31,31,30, 31, 30, 31/
c
c...leap year correction
      Leap = 0
      if(MOD(Iyr,4) .eq. 0) Leap = 1
c
      if(Iday .lt. 32 ) then

```

```

        I = 1
        Month = 1
        Mday = Iday
        NN = Nmonth(Month)
        go to 999
    endif
c
    do 10 I = 2, 12
        if(Iday .lt. Mth(I)+Leap) go to 11
    10 continue
c
    11 Month = I
        if(Month .gt. 2) then
            Mday = Iday - Mth(Month-1) - leap + 1
            NN = Nmonth(Month)
        else
            Mday = Iday - Mth(Month-1) + 1
            NN = Nmonth(Month) + Leap
        endif
c
c
    999 return
    end
c
c*****
c      subroutine blocker
c
c      parameter(N=31, M=288)
c
c...md(i)    number of measurements on day i
c...NN       number of days in the current month
c...nb       number of blocks per characteristic
c...ibc      max number of characters per block
c...nc       number of characters per block
c...ldb      counter for number of days/block
c...Ndb(15)  number of days in each block to be written, 15 would
c...         support a 2 minute measurement schedule
c
c...times and measurements
        Common/times/md(N),Ihv(N,24),NN,nb,Ndb(15)
c
        ibc = 9000
        nc = 0

```

```

c...preset
  nb = 1
  ldb = 0
  do 5 i=1,15
5  Ndb(i) = 0
  Ndb(1) = 31
c
  do 20 i=1,NN
c...six characters per measurement
  ldb = ldb + 1
  nc = nc + md(i)*6
  if(nc .gt. ibc) then
    Ndb(nb) = ldb - 1
    ldb = 1
    nb=nb + 1
    nc = md(i)*6
  endif
20 continue
c...when more than one block remainder goes in last block
  if(nb .gt. 1) Ndb(nb) = ldb
c
  write(4,*) 'nb, Ndb'
  write(4,'(I4,5x,15I4,/)') nb, Ndb
  return
end

c
c*****
  subroutine preset
c
  parameter(N=31, M=288)
c
  Integer*2
cfoF2,cfoF1,cM3000,cMUF,cfmin,cfoEs,cfminF,cfminE,cfoE,
+ cfxI,chpF,chpF2,chpE,chpEs
c
  Common/Chard1/cfoF2(N,M),cMUF(N,M),cfoF1(N,M)
  Common/chard2/cM3000(N,M),cfmin(N,M),cfoEs(N,M)
  Common/chard3/cfminF(N,M),cfminE(N,M),cfoE(N,M)
  Common/chard4/cfxI(N,M),chpF(N,M),chpF2(N,M)
  Common/chard5/chpE(N,M),chpEs(N,M)
c
c...times and measurements
  Common/times1/md(N),Ihv(N,24),NN,nb,Ndb(15)

```

```

      Integer*4 Itimes
      Common/time4/Itimes(N,M)
c
      do 3 k=1,24
      do 3 i=1,N
      Ihv(i,k) = 0
3 md(i) = 0
c
      do 10 i=1,N
      do 10 j=1,M
      Itimes(i,j) = 0
      cfoF2(i,j) = 9999
      cfoF1(i,j) = 9999
      cM3000(i,j) = 9999
      cMUF(i,j) = 9999
      cfmin(i,j) = 9999
      cfoEs(i,j) = 9999
      cfminF(i,j) = 9999
      cfminE(i,j) = 9999
      cfoE(i,j) = 9999
      cfxI(i,j) = 9999
      chpF(i,j) = 9999
      chpF2(i,j) = 9999
      chpE(i,j) = 9999
10 chpEs(i,j) = 9999
c
      return
      end
C
c*****
c....read 1990 ADEP IIWG suggested format
      Subroutine
      RDADEP(EOF,IU,IDF1,IPREF,SCALED,GCONST,IAF,DTT,IOTF,
      +
      IOTHF,IOAF,IODF,FOF,IOTF1,IOTHF1,IOAF1,IODF1,FOF1,IOTE,IOTHE,
      +
      IOAE,IODE,FOE,IXTF,IXAF,IXDF,FXF,IXTF1,IXAF1,IXDF1,FXF1,IXTE,
      +
      IXAE,IXDE,FXE,MEDF,MEDE,MEDES,THF2,THF1,THE,THVAL,IedF,Sysdes)
C
C      This subroutine reads scaled Ionospheric data from a textfile
C      in the format specified by the University Of Lowell Center for

```

```

C      Atmospheric Research for ionospheric data produced by
Digisonde
C      256 digital ionospheric sounders and processed by the ADEP
C      ARTIST Data Editing Program. The format is that suggested by
the
C      IIWG Workshop -- ADEP 90 --
C      The EOF variable is set if an End Of File is encountered during
C      the file read.
C
C      For further information, see the ULowell document:
C      ARTIST Data Editing Program Output Format (02Apr88)
C      NOTE:  Values which are all conatined on one line are not
C              read in implied DO loops so as to minimize the effect
C              of errors on the subsiquent records.
C
C      Reads data from the FORTRAN unit number IU into the data
arrays.
C
      LOGICAL EOF
      INTEGER I,IU
      INTEGER
IDFI(80),IPREF(60),IAF(20),IOTF(400),IOTHF(400),IOAF(400),
+
IODF(400),IOTF1(150),IOTHF1(150),IOAF1(150),IODF1(150),IOTE(150
)
+
,IOTHE(150),IOAE(150),IODE(150),IXTF(400),IXAF(400),IXDF(400),
+      IXTF1(150),IXAF1(150),IXDF1(150),IXTE(150),IXAE(150),
+      IXDE(150),MEDF(20),MEDE(20),MEDES(20),ledF(120)
      REAL
SCALED(30),GCONST(5),DTT(16),FOF(400),FOF1(150),FOE(150),
+      FXF(400),FXF1(150),FXE(150),THF2(20),THF1(20),THE(20),
+      THMON(20),THVAL(20)
      Character*120 Sysdes
C
c...formats
101  FORMAT (40I3)
102  FORMAT (16F7.3)
103  Format(A120)
104  FORMAT (120Z1)
105  FORMAT (15F8.3)
106  FORMAT (60I2)
107  FORMAT (120I1)

```

```

108 FORMAT (20F6.3)
109 FORMAT (13E9.4E1)
c...data file index
C   The data file index integers should all be on one line.
    READ(IU,101,END=9) IDFI
C
C...Geophysical constants -- Code 1
    IF(IDFI(1).GT.0) READ(IU,102,END=9) GCONST
c
C...System description -- Code 2
    if(IDFI(2).gt.0) READ(IU,103,END=9) Sysdes
c
C...ionogram sounding settings (preface) -- Code 3
    if(IDFI(3).gt.0) READ(IU,104,END=9) (IPREF(I),I=1,IDFI(3))
C
C...scaled ionogram parameters
    IF(IDFI(4).GT.0) READ(IU,105,END=9) (SCALED(I),I=1,IDFI(4))
C
C...ARTIST analysis flags
    IF(IDFI(5) .GT.0) READ(IU,106,END=9) IAF
C
C...Doppler translation table
    IF(IDFI(6).GT.0) READ(IU,102,END=9) DTT
C
C...O-trace F2 points
c...virtual height
    IF(IDFI(7).GT.0) READ(IU,101,END=9) (IOTF(I),I=1,IDFI(7))
c
c...true height
    IF(IDFI(8).GT.0) READ(IU,101,END=9) (IOTfF(I),I=1,IDFI(8))
c
c...Amplitudes
    IF(IDFI(9).GT.0) READ(IU,106,END=9) (IOAF(I),I=1,IDFI(9))
c
c...Doppler numbers
    IF(IDFI(10).GT.0) READ(IU,107,END=9) (IODF(I),I=1,IDFI(10))
c
c...Frequency table
    IF(IDFI(11).GT.0) READ(IU,108,END=9) (FOF(I), I=1,IDFI(11))
C
C...O-trace F1 points
c...virtual height
    IF(IDFI(12).GT.0) READ(IU,101,END=9) (IOTF1(I),I=1,IDFI(12))

```

```

c
c...true height
      IF(IDFI(13).GT.0) READ(IU,101,END=9) (IOTf1(I),I=1,IDFI(13))
c
c...Amplitudes
      IF(IDFI(14).GT.0) READ(IU,106,END=9) (IOAF1(I),I=1,IDFI(14))
c
c...Doppler number
      IF(IDFI(15).GT.0) READ(IU,107,END=9) (IODF1(I),I=1,IDFI(15))
c
c...Frequency table
      IF(IDFI(16).GT.0) READ(IU,108,END=9) (FOF1(I), I=1,IDFI(16))
C
C...O-trace E points
c...virtual heights
      IF(IDFI(17).GT.0) READ(IU,101,END=9) (IOTE(I),I=1,IDFI(17))
c
c...true height
      IF(IDFI(18).GT.0) READ(IU,101,END=9) (IOTfE(I),I=1,IDFI(18))
c
c...Amplitudes
      IF(IDFI(19).GT.0) READ(IU,106,END=9) (IOAE(I),I=1,IDFI(19))
c
c...Doppler numbers
      IF(IDFI(20).GT.0) READ(IU,107,END=9) (IODE(I),I=1,IDFI(20))
c
c...Frequency table
      IF(IDFI(21).GT.0) READ(IU,108,END=9) (FOE(I), I=1,IDFI(21))
C
C...X-trace F2 points
c...virtual heights
      IF(IDFI(22).GT.0) READ(IU,101,END=9) (IXTF(I),I=1,IDFI(22))
c
c...Amplitudes
      IF(IDFI(23).GT.0) READ(IU,106,END=9) (IXAF(I),I=1,IDFI(23))
c
c...Doppler numbers
      IF(IDFI(24).GT.0) READ(IU,107,END=9) (IXDF(I),I=1,IDFI(24))
c
c...Frequency table
      IF(IDFI(25).GT.0) READ(IU,108,END=9) (FXF(I),I=1,IDFI(25))
C
C...X-trace F1 points

```

```

c...virtual heights
    IF(IDFI(26).GT.0) READ(IU,101,END=9) (IXTF1(I),I=1,IDFI(26))
c
c...Amplitudes
    IF(IDFI(27).GT.0) READ(IU,106,END=9) (IXAF1(I),I=1,IDFI(27))
c
c...Doppler numbers
    IF(IDFI(28).GT.0) READ(IU,107,END=9) (IXDF1(I),I=1,IDFI(28))
c
c...Frequency table
    IF(IDFI(29).GT.0) READ(IU,108,END=9) (FXF1(I),I=1,IDFI(29))
C
C...X-trace E points
c...virtual heights
    IF(IDFI(30).GT.0) READ(IU,101,END=9) (IXTE(I),I=1,IDFI(30))
c
c...Amplitudes
    IF(IDFI(31).GT.0) READ(IU,106,END=9) (IXAE(I),I=1,IDFI(31))
c
c...Doppler numbers
    IF(IDFI(32).GT.0) READ(IU,107,END=9) (IXDE(I),I=1,IDFI(32))
c
c...Frequency table
    IF(IDFI(33).GT.0) READ(IU,108,END=9) (FXE(I),I=1,IDFI(33))
C
C...Median amplitude of F echo
    IF(IDFI(34).GT.0) READ(IU,106,END=9) (MEDF(I),I=1,IDFI(34))
C...Median amplitude of E echo
    IF(IDFI(35).GT.0) READ(IU,106,END=9) (MEDE(I),I=1,IDFI(35))
C...Median amplitude of Es echo
    IF(IDFI(36).GT.0) READ(IU,106,END=9) (MEDES(I),I=1,IDFI(36))
C
C...F2 layer true height parameters
    IF(IDFI(37).GT.0) READ(IU,109,END=9) (THF2(I),I=1,IDFI(37))
c
C...F1 layer true height parameters
    IF(IDFI(38).GT.0) READ(IU,109,END=9) (THF1(I),I=1,IDFI(38))
C...E layer true height parameters
    IF(IDFI(39).GT.0) READ(IU,109,END=9) (THE(I),I=1,IDFI(39))
C
C...Valley parameters
    IF(IDFI(40).GT.0) READ(IU,109,END=9) (THVAL(I),I=1,IDFI(40))
C

```

c...Edit Flags

IF(IDFI(41).GT.0) READ(IU,107,END=9) (ledF(I),I=1,IDFI(41))

c

EOF = .FALSE.

RETURN

9 EOF = .TRUE.

RETURN

END

c

APPENDIX B  
PROGRAM RD-CHARS.FOR

Storage: 2

c

PROGRAM Rd\_Chars

C

c...written by Bob Gamache

C Reads database characteristics files

C in IIWG Workshop format.

c...last modified 13 December,1990 by Bob Gamache

C

parameter(N=31, M=288)

c

CHARACTER\*40 IFILE,Iout

CHARACTER\*30 Station,System

CHARACTER\*10 Scale>Edit

CHARACTER\*5 cStat,Ch5(9000),C5(24)

CHARACTER\*6 Ch6(9000)

CHARACTER\*10 char(80),units(80),UT

character\*2 code(80)

dimension Icount(24),IRange(24)

c

c...times and measurements

Common/tm/md(N),NN,nmt

c

WRITE(\*,\*) ' Enter name of input file.'

READ(\*,'(A)') IFILE

OPEN(1,FILE=IFILE,FORM='FORMATTED',

+ ACCESS='SEQUENTIAL',RECL=9000)

c

WRITE(\*,\*) 'Enter name of output file.'

READ(\*,'(A)') Iout

OPEN(2,FILE=Iout,FORM='FORMATTED',

+ ACCESS='SEQUENTIAL', RECL=9000)

c

c...read/write first record

read(1,'(A30,A5,A4,2F5.1,2A10,A30)',end=500) station,cStat,UT,

+ Rlat,Rlong,Scale>Edit,System

read(1,'(30I4)') Iyr,Month,NN,ncar,nmt,md

read(1,'(12A10)') (Char(L),L=1,ncar)

read(1,'(12A10)') (units(L),L=1,ncar)

read(1,'(60A2)') (code(L),L=1,ncar)

c

c

write(2,'(A30,5x,A5,a,f5.1,a,f5.1,5x,a,A4)') Station,cStat,

```

+ ' latitude:',Rlat,' longitude:',Rlong,' UT:',UT
  write(2,'(1x,a10,5x,a10,2x,a,A30,a,I2,1x,i4)')Scale>Edit,
+ ' System: ',System,' Month: ',Month,Iyr
c
  Write(2,'(/,I5,a,I5,a)') NN,' Days ',nmt,' # measurements total'
  Write(2,'(/,a)') ' # measurements per day'
  Write(2,'(30I4)') md
c
c
  write(2,'(/,a,/)') measurement times: hours:minutes:seconds'
c...output measurement times
c...number of measurement lines
  nml = nmt/20
  if( (nmt-nml*20) .gt. 0) nml = nml + 1
c...number in last line
  nll = nmt - nml*20
c...read 20 words at one time
  ns = 1
  ne = ns + 19
c
  do 20 L=1, nml
    read(1,'(20A6)') (Ch6(kk),kk=ns,ne)
    ns = ns + 20
    ne = ns + 19
c...last line
  if(l .eq. nml) ne = ns + nll - 1
  20 continue
  write(2,'(15(2x,A6))') (CH6(j), j=1, nmt)
c
c...output characteristics
c
  do 40 Nc=1, ncar
    write(2,'(/,a,a10,1x,a10,2x,a11,a2/)') characteristic: ',
+ char(Nc),units(Nc),' URSI code ',code(Nc)
c
c...number of measurement lines for characteristics
  nml = nmt/24
  if( (nmt-nml*24) .gt. 0) nml = nml + 1
c...number in last line
  nll = nmt - nml*24
c...read 24 words at one time
  ns = 1
  ne = ns + 23

```

```

c
  do 30 L=1, nml
    read(1,'(24A5)') (Ch5(kk),kk=ns,ne)
    ns = ns + 24
    ne = ns + 23
c...last line
    if(l .eq. nml) ne = ns + nll - 1
  30 continue
    write(2,'(15(3x,A5))') (Ch5(j), j=1, nmt)
c
c...medians
    read(1,'(24A5)') (C5(L),L=1,24)
    write(2,'(/,a,/)'      medians'
    write(2,'(12(3x,A5))') (C5(J), J=1, 24)
c
c...median counts and Range
    read(1,'(24(I2,I3))') (Icount(L),IRange(L),L=1,24)
    write(2,'(/,a,/)'      counts'
    write(2,'(12(3x,I2))') (Icount(J), J=1, 24)
c
c...Range
    write(2,'(/,a,/)'      range'
    write(2,'(12(3x,I3))') (Irange(J), J=1, 24)
c
c...Upper Quartiles
    read(1,'(24A5)') (C5(L),L=1,24)
    write(2,'(/,a,/)'      upper quartiles'
    write(2,'(12(3x,A5))') (C5(J), J=1, 24)
c
c...Lower Quartiles
    read(1,'(24A5)') (C5(L),L=1,24)
    write(2,'(/,a,/)'      lower quartiles'
    write(2,'(12(3x,A5))') (C5(J), J=1, 24)
c
c...Upper Deciles
    read(1,'(24A5)') (C5(L),L=1,24)
    write(2,'(/,a,/)'      upper deciles'
    write(2,'(12(3x,A5))') (C5(J), J=1, 24)
c
c...Lower Deciles
    read(1,'(24A5)') (C5(L),L=1,24)
    write(2,'(/,a,/)'      lower deciles'
    write(2,'(12(3x,A5))') (C5(J), J=1, 24)

```

c

40 continue

c

500 continue

end